



A Dynamic Bayesian Intelligent Interface Agent

Proceedings of the Sixth International Interfaces Conference (Interfaces 97), pp. 118-120

Montpellier, France, May 1997

Scott M. Brown, Eugene Santos Jr., and Sheila B. Banks

Department of Electrical & Computer Engineering
Air Force Institute of Technology
2950 P Street, Wright-Patterson AFB, OH 45433-7765

The views expressed in this article are those of the authors and do not reflect the official policy or position of the United States Air Force, Department of Defense, or the US Government

Approved for public release; distribution unlimited.

A Dynamic Bayesian Intelligent Interface Agent

Scott M. Brown and Eugene Santos Jr. and Sheila B. Banks

Department of Electrical and Computer Engineering

Air Force Institute of Technology

Wright-Patterson AFB, OH 45433-7765

{sbrown, esantos, sbanks}@afit.af.mil

December 18, 1997

Abstract

This paper presents an intelligent user interface agent architecture based on Bayesian networks. Using a Bayesian network not only dynamically captures and models user behavior, but it also dynamically captures and models uncertainty in the interface's reasoning process. We present an approach that allows our intelligent interface agent (IIA) to alter its own topology to better adapt itself to modeling a particular user. We present several metrics that provide useful information concerning the performance of our IIA. IIA's sound semantics and mathematical basis enhances its ability to make correct, intelligent inferences as to the user's needs.

KEYWORDS: intelligent user interface, Bayesian network, intelligent agent, generic expert system, knowledge representation, reinforcement learning, cognitive reasoning

Methods, Models, Tools, and Techniques – Oral Presentation

A Dynamic Bayesian Intelligent Interface Agent

Scott M. Brown and Eugene Santos Jr. and Sheila B. Banks

Department of Electrical and Computer Engineering

Air Force Institute of Technology

Wright-Patterson AFB, OH 45433-7765

{sbrown, esantos, sbanks}@afit.af.mil

1 Introduction

Intelligent user interface (UI) research is primarily focused on human-computer interface issues, especially with the abilities and usability of interfaces. However, intelligent interface researchers have put little emphasis on improving the structures representing the intelligence of these interfaces. In this paper, we are primarily concerned with presenting the utilization of Bayesian networks in intelligent user interfaces.

GESIA (Generic Expert System Intelligent Assistant) [6] is an intelligent user interface architecture conceived out of the development of a generic expert system. This expert system, called PESKI (Probabilities, Expert Systems, Knowledge, and Inference) [11], is a collection of expert system tools under one architecture that is designed to be totally independent of any application domain, i.e., generic. The tools contained in PESKI include an inference engine, a knowledge base (called a Bayesian knowledge base [1] or BKB), a knowledge acquisition tool, a knowledge base verification and validation tool, and a data mining tool.

It is widely agreed that basing decisions on an accurate cognitive model of the user is important for effective prediction of user intent and that the interface should be able to collect and model information about false inferences [9, 12]. Collecting such data is cognitively and computationally difficult. DeWitt notes that not all naturalistic (i.e., observable) properties play an interesting role in a user's causal model [3]. That is, only certain observable actions and information in a user's "world" will have relevance to that user. Therefore, to effectively and efficiently capture user intent, our model should not attempt to model every possible action the user may exhibit, but only those that are relevant, i.e., most likely to be exhibited.

Many research interfaces use rule-based intelligence [4]. Rule-based representations, like those used in most intelligent user interfaces, fail in two key areas - representing uncertainty and dynamic user modeling. The use of "probability modules" [13] is an ad hoc approach to determining answer reliability, i.e., uncertainty. Furthermore, the addition and deletion of rules to dynamically model a user is ad hoc. Therefore, knowledge representations that can dynamically capture and model uncertainty in human-computer interaction can improve the modeling of the user and user interface states in an intelligent user interface. One knowledge representation that is ideal for representing uncertainty is a Bayesian Network (BN). A Bayesian network is a mathematically correct and semantically sound model for representing uncertainty that provides a means to show probabilistic relationships between items [10].

This paper is organized as follows. In Section 2, we present the Bayesian network knowledge representation utilized by our user interface intelligent agent called the Intelligent Interface Agent. In Section 3, we present our methodology for allowing the Bayesian networks change *dynamically* as the user interacts with the system and provide several metrics for determining the applicability of an approximation and intelligent agent's performance on representing a user. Finally, we discuss future research and our conclusions in Section 4.

2 GESIA Development

The goals of GESIA's development are threefold:

- To provide for user access to the many tools of the expert system using proven user interface design theory and implementation methods.
- To maintain the domain independence of the expert system, or, in other words, ensure the expert system can easily be transplanted from one application domain to another.
- To assist the user with managing the complexities of the generic expert system through the use of intelligence or reasoning capability.

GESIA's IUI provides access to the generic expert system tools and applications. There are three major layers of the architecture: the graphical layer, the intelligent interface agent (IIA) layer, and the system layer.

The graphical layer of the IUI contains Motif/OSF standard interface widgets. Together, these widgets form the visual part of the interaction between the user and the expert system applications. The system layer provides the link between the IUI and the expert system applications through a series of tool drivers, one for each expert system application tool.

The IIA layer is the most complex and important layer of the IUI. This layer controls the communications and intelligence aspects of the interface and is composed of three sublayers: the adaptation layer, the adaptive layer, and the communications layer. The adaptation layer manages and tracks all adaptations the user makes to the IUI. The adaptive layer communicates directly with the interface learning network gateway (explained in detail in the next section) to perform interface initiated adaptations to the IUI based on perceived user behavior. Finally, the communications layer controls the various modes of communication available to the interface such as structured text, graphical manipulation, and natural language.

2.1 The GESIA Interface Learning Network

The GESIA Interface Learning Network (ILN) is the heart of the IIA layer. The Bayesian network knowledge representation captures, stores, and models user and interface behavior. The network is composed of two semantically different nodes: interface learning nodes and interface information nodes. The network is also composed of containers that store learned user and user class behavior data and a network communications gateway.

2.1.1 Interface Learning Node

Semantically, the interface learning node represents behavior that the interface has collected about a particular system user or class of users. This node is named according to the behavior collected, for example "User Prefers Knowledge Acquisition" or "User's Class Prefers Knowledge Acquisition." Each node's probability is stored as a fraction. The denominator of the fraction represents the number of learning occurrences that affect the node. The numerator of the fraction represents the number of learning occurrences that add to the truthfulness of the node (i.e., a higher probability).

After the node is instantiated, the interface learning network loads stored data about the current system user into the interface learning node. Whenever the system user exhibits behavior represented by the node, the interface will call the node's update method to record the behavior. The updating is based on simple reinforcement learning.

2.1.2 Interface Information Node

Semantically, the interface information node represents a possible user state. Each interface information node is supported by two or more interface learning nodes and zero or more interface information nodes. When an interface information node is instantiated, it receives and stores access information to all its child nodes. The node sits “dormant” until the interface queries it for its probability, in order to make inferences as to the user’s future state (i.e., action). When the node is queried, this node combines the probabilities of all the supporting nodes to determine the probability that state is true using Bayes Theorem [10]. This value represents the probability that the node’s state is true. This node is named after the state it represents, for example, “User is Using Graphical Communication.”

3 Dynamic Interface Learning Networks

The networks used and presented previously [5, 6] have been pedagogical examples. While although they represent the concepts and advantages of using Bayesian networks as a knowledge representation in intelligent user interfaces, they represent only a microcosm of the entire system and the possible actions a user may perform while using the system. A simple minded approach to creating an interface learning network to represent the user’s actions is to have a single node for each action possible in the system. This ILN could potentially have thousands of interface information nodes and approximately as many interface learning nodes. Surely, this would provide the most accurate representation of the system and the possible actions a user could perform at any time. Creating a node for every possible action in our system and allowing this node’s dependencies (i.e., parent nodes) to be connected to it allows us to represent our system exactly. However, there are two main problems with this approach. First, since our user will rarely if ever exhibit certain actions, the probability of certain nodes will be very small. When we combine our probabilities, these “irrelevant” probabilities can have the effect of ignoring the relevant node’s probability. In Dewitt’s terms, these actions are not “causally efficacious”. Secondly, it is well known that belief propagation is NP-hard [2]. Therefore, we must find an approximation to our network that models only relevant nodes and is a good tradeoff between computational complexity and representation exactness. This implies a dynamic ILN structure, where we add and remove relevant nodes in our ILN. How to determine what nodes are relevant can be difficult.

3.1 Methodology

As mentioned previously, there are computational limits of modeling the entire user interface, i.e., every possible action the user may perform, as an ILN. However, we take solace in the fact that for any given user, that user will only display a subset of all possible actions during a given session with the system. Yet, the subset of possible actions may be too large to use as a basis of a complete interface learning network. Therefore, we must restrict our interface learning network further.

There has been much research in the field of approximating Bayesian Networks [10, 7]. Current techniques revolve around stochastic simulation, Likelihood Weighting, and Logic Sampling. Since a user will only exhibit a subset of all possible actions, we only allow a total of the N most relevant nodes to be present at any time in our network. When a user performs an action, this action is communicated via the ILN gateway to the IIA. This action may or may not be represented in the ILN. If it is, we update the network and calculate the new probabilities. If it is not, we modify the existing ILN topology. Currently, we add

the new node, calculate the new probabilities, and then remove the least relevant node, i.e., lowest priority. In this way, we can view our ILN as a priority queue of user actions, where the highest priority actions are most relevant.

3.2 Metrics

We have chosen a method for dynamically changing a user’s ILN. We are concerned with finding the method to represent a particular user. Are the resulting ILNs truly representative of our user? We define several objective metrics (versus subjective usability comparisons) that give us insight into the performance of our ILN.

Absolute thrashing is the result of an observable property (i.e., node) repeatedly entering and leaving the relevancy set. We are concerned with adding a node to the interface learning network only to have it never queried and leave the network some time later. *User thrashing* occurs when a user’s intent and the system’s measure of intent, as represented by the interface learning network, swings from one extreme to another. If we are truly modeling user intent, we should desire to capture user thrashing. However, we also desire to avoid thrashing of the system’s measure of user intent so we can make accurate predictions of the user’s intent. As a concrete example, consider an aunt who is known to have drastic mood swings. You are aware of this, and develop ways of observing her current behavior to find a promising way of approaching her. You never vary your approach drastically. *Rate of divergence* is a measure of how quickly a node leaves the interface learning network. We are concerned with ensuring a node that was added to our network, but is not used often will exit the network quickly, therefore allowing more useful nodes in the relevancy set. *Class thrashing* is the result of a user who is diametrically opposed to his/her user class and therefore the network initially does not represent a user well and consequently, must “learn back” a user’s behavior. This type of thrashing not only affects the user by making incorrect inferences, but affects the user class the misplaced user is currently a member of. *Rate of convergence* is the speed the “momentum” of past observed behavior is overcome by changes in current behavior and therefore, how quickly the probability of a node will settle out to a particular value within an epsilon and is important in conjunction with class thrashing. We desire to know how fast a network will allow a user to overcome past behavior. For example, a user may exhibit a particular behavior for a “long” time and then suddenly change behavior, perhaps as the result of some new stimuli in the user’s environment. A fast rate of convergence will quickly allow the user model to overcome the past behavior and accurately model the current behavior.

4 Conclusions

The prototype interface learning network used to date [5] needs to be expanded to capture additional actions. Expansion of the number of actions GESIA will monitor allows a more accurate model of the user’s behavior to be maintained. Our current IIA uses a dynamic “hand-coded” interface learning network. We determine *a priori* the actions we will monitor. This is not unlike Maes’ hand-coded situations [8]. This *a priori* determination limits the number of user actions we must monitor in our system. While although most “hand-coded” user models are static, ours allows the dynamic addition and deletion of nodes associated with a particular Bayesian knowledge base. We limit the number of BKB associated nodes allowed in the user’s interface learning network at any one time. If the user loads a BKB that is not represented in the current network, we add it to the network. If we have reached our network size limitation (currently set at a hard limit of five BKB nodes), we delete the lowest

probability node from the network. In this way, the most relevant (i.e., highest probability) nodes are in our network at any point in time.

The intelligence of the interface can be enhanced if the interface is able to interpret why it makes bad predictions. We propose a dynamic “meta-level” of inferencing, capable of modifying the user’s interface learning network topology as the user performs action in PESKI. To realize this, we must be able to determine “real-time” what is happening with the user. Most usability studies are done “off-line” and have no immediate bearing on the user model. The incorporation of temporal reasoning into this representation would allow the interface to predict user traits based on the patterns [14].

The ability of this network to model user behavior can be expanded by designing the interface to understand the user behavior. For example, if the interface measures patterns of indicator swings the interface may begin to classify these patterns. The interface may then be able to assign patterns to user traits, such as moods. The incorporation of temporal reasoning into this representation would allow the interface to predict user traits based on the patterns.

The intelligence of the interface can be enhanced if the interface is able to interpret why it makes bad predictions. The metrics presented can be used to determine that a problem exists in our ILN’s representation of a user’s intent and find ways to solve the problem.

The interface learning network provides GESIA with an effective knowledge representation for user, user class, and interface behavior. The use of Bayesian networks over rule-based systems to accurately model the user better captures the uncertainty of user actions by using sound semantics and a firm mathematical basis. Initial tests show noticable savings in the user’s physical workload while accurately predicting users’ behavior. For our architecture to be truly helpful, we have designed it as a dynamic universal agent, capable of being used with any user interface willing to “speak” the common language presented here. For any IIA, we have presented several metrics that provide an insight into the performance of our network. Furthermore, the momentum of learned behavior in one direction can be reversed and changed to another direction of behavior quickly.

References

- [1] Darwyn O. Banks. Acquiring consistent knowledge for bayesian forests. Master’s thesis, Graduate School of Engineering, Air Force Institute of Technology, Wright-Patterson AFB OH, 1995.
- [2] Gregory F. Cooper. The computational complexity of probabilistic inference using bayesian belief networks. *Artificial Intelligence*, 42:393–405, 1990.
- [3] Richard DeWitt. Vagueness, semantics, and the language of thought. In *Psyche*, volume 1. December 1993.
- [4] Avelino J. Gonzalez and Douglas D. Dankel. *The Engineering of Knowledge-Based Systems*. Prentice Hall, Englewood Cliffs, NJ, 1993.
- [5] Robert A. Harrington. Utilizing bayesian techniques for user interface intelligence. Master’s thesis, Air Force Institute of Technology, Wright-Patterson AFB, OH, 1996.
- [6] Robert A. Harrington, Sheila Banks, and Eugene Santos, Jr. Development of an intelligent user interface for a generic expert system. In Michael Gasser, editor, *Online Proceedings of the 1996 Midwest Artificial Intelligence and Cognitive Science Conference*, 1996. URL: <http://www.cs.indiana.edu/event/maics96/Proceedings/harrington.html>.
- [7] Eugene Santos Jr., Solomon Eyal Shimony, and Edward Williams. Hybrid algorithms for approximate belief updating in bayes nets. *to appear in International Journal of Approximate Reasoning*, 1997.

- [8] Pattie Maes. Agents that reduce work and information overload. *Communications of the ACM*, 37(7):811–821, July 1994.
- [9] Reinhard Oppermann. Adaptively supported adaptivity. *International Journal of Human-Computer Studies*, 40:455–472, 1994.
- [10] Judea Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann, San Mateo, CA, 1988.
- [11] Eugene Santos, Jr. A fully integrated probabilistic framework for expert system development. Technical report, Department of Electrical and Computer Engineering, Air Force Institute of Technology, Wright-Patterson AFB, OH, 1993.
- [12] C. G. Thomas. Design, implementation and evaluation of an adaptive user interface. *Knowledge-Based Systems*, 6:230–238, 1993.
- [13] Patrick Henry Winston. *Artificial Intelligence, 2d Ed.* Addison-Wesley, 1984.
- [14] Joel D. Young and Eugene Santos, Jr. Introduction to temporal bayesian networks. In Michael Gasser, editor, *Online Proceedings of the 1996 Midwest Artificial Intelligence and Cognitive Science Conference*, 1996. URL: <http://www.cs.indiana.edu/event/maics96/Proceedings/Young/maics-96.html>.